SNAP: Towards Segmenting Anything in Any Point Cloud

Supplementary Material

This supplementary document is structured as follows:

7. Model Details

7.1. Model Architecture

To complement the simplified pipeline presented in the main paper, we provide a more detailed illustration of SNAP in **Fig. 6**. Specifically, the Mask Decoder is decomposed into a Prompt-Point Attention module and three dedicated MLP heads that process the mask, score, and CLIP tokens, together with the spatial prompt embeddings. For completeness, we also indicate the corresponding supervision signals, showing how each type of prediction contributes to the overall training objective through its associated loss functions.

7.2. Details on Loss Functions

Focal and Dice Loss. Following Interactive4D [5], we apply a distance-based, click-localized weight to the point-wise Focal loss and Dice loss terms ($\mathcal{L}_{\text{focal}}$ and $\mathcal{L}_{\text{dice}}$). Concretely, for each point $\mathbf{p}_i \in \mathbf{P}$, we compute its normalized distance to its nearest spatial prompt point $\mathbf{p}^*_{\text{sp}}$: $d_i = \text{Dist}(\mathbf{p}_i, \mathbf{p}^*_{\text{sp}})$. If d_i is below a threshold τ_d, the weight is defined to decay linearly from w_{max} to w_{min} as the distance increases; otherwise, the weight is set to w_{min}. Formally, the weight of each point is defined as:

$$w(\mathbf{p}_i) = \begin{cases} w_{\max} - (w_{\max} - w_{\min})\, d_i, & d_i < \tau_d, \\ w_{\min}, & \text{otherwise.} \end{cases} \tag{3}$$

In our implementation, we set $w_{\text{max}} = 2$, $w_{\text{min}} = 1$, and $\tau = 0.5$. This weighting strategy increases the contribution of points closer to spatial prompts (clicks), encouraging the model to focus its supervision around click regions while preserving global mask consistency.

Auxiliary Loss. In addition to the final mask prediction loss described above, we strengthen supervision by leveraging the spatial prompt embeddings corresponding to individual clicks. The intuition is that each click should independently guide a plausible mask prediction, rather than only contributing through the aggregated mask token. To achieve this, we treat the P prompt embeddings extracted from $\mathbf{Z}\text{sp} \in \mathbb{R}^{M\times(P+3)\times D}$ as *auxiliary mask tokens* and feed them through the same mask head described in the method section. This yields $M \times P$ auxiliary mask predictions, which are supervised using standard point-wise focal and Dice loss terms. The resulting auxiliary loss $\mathcal{L}_{\text{aux}}$ encourages individual clicks to directly align with the segmentation masks, thereby providing more fine-grained supervision.

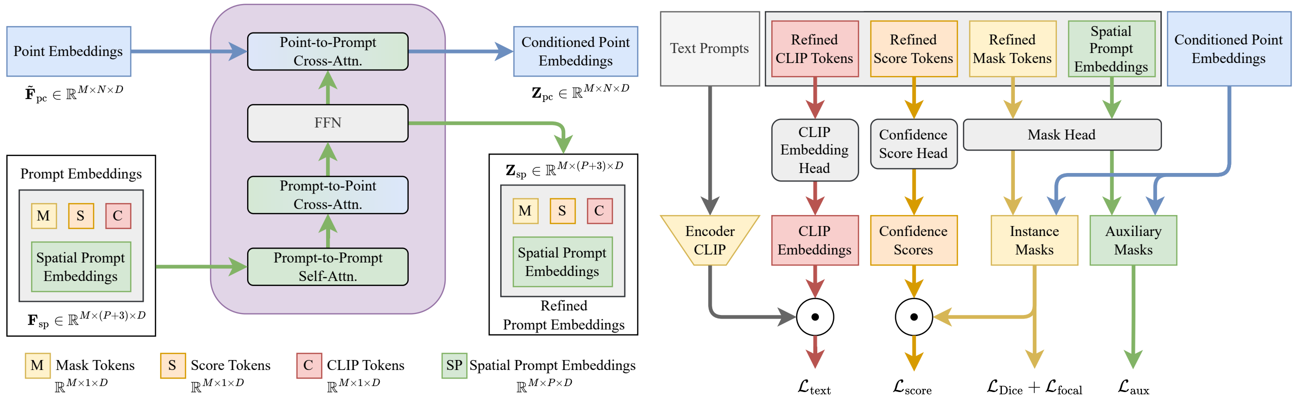
Confidence Score Loss. To improve the reliability of mask confidence score estimation, we supervise this score prediction process with $\mathcal{L}_{\text{score}}$. Intuitively, this score should reflect the quality of the predicted mask, which we approximate by its intersection-over-union (IoU) with its corresponding ground-truth mask. Concretely, given the predicted mask $\mathcal{M}_i$, we first obtain a binarized mask by thresholding it: $\mathcal{M}_i > \tau$. The IoU between this mask and its ground-truth counterpart $\mathcal{M}^*_i$ is then computed and used as the regression target: $\mathcal{S}^*_i = \text{IoU}(\mathcal{M}_i > \tau, \mathcal{M}^*_i)$. Finally, we formulate the score loss as a mean squared error (MSE) between the predicted score and the IoU target:

$$\mathcal{L}_{\text{score}} = \frac{1}{M}\sum_{i=1}^{M}\left(\mathcal{S}_i - \mathcal{S}^*_i\right)^2, \tag{4}$$

where $\mathcal{S}_i$ denotes the predicted score for the i-th mask.

Text Loss. To supervise the predicted CLIP tokens $\mathbf{L}_{\text{CLIP}}$, we follow a prototype-based classification scheme against the CLIP text vocabulary embeddings $\mathbf{T} \in \mathbb{R}^{C\times D_{\text{CLIP}}}$. After L2 normalization, cosine similarities between $\mathbf{L}_{\text{CLIP}}$ and $\mathbf{T}$ yield logits $\mathbf{Z} = \mathbf{L}_{\text{CLIP}} \cdot \mathbf{T}^\top \in \mathbb{R}^{M\times C}$. For the i-th sample with ground-truth label y, let $p_i = \text{softmax}(\mathbf{Z}_i)_y$ denote the probability of the correct class. We then apply a focal loss with focusing parameter $\gamma = 2.0$ and no additional class re-weighting, which gives the text loss:

$$\mathcal{L}_{\text{text}} = \frac{1}{M}\sum_{i=1}^{M}\left(1 - p_i\right)^{\gamma}\left(-\log p_i\right). \tag{5}$$


Point Embeddings
$\tilde{\mathbf{F}}_{\text{pc}} \in \mathbb{R}^{M\times N\times D}$
Point-to-Prompt Cross-Attn.
FFN
Prompt-to-Point Cross-Attn.
Prompt-to-Prompt Self-Attn.
Conditioned Point Embeddings
$\mathbf{Z}_{\text{pc}} \in \mathbb{R}^{M\times N\times D}$
Prompt Embeddings
M
S
C
Spatial Prompt Embeddings
$\mathbf{F}_{\text{sp}} \in \mathbb{R}^{M\times (P+3)\times D}$
$\mathbf{Z}_{\text{sp}} \in \mathbb{R}^{M\times (P+3)\times D}$
Refined Prompt Embeddings
M Mask Tokens $\mathbb{R}^{M\times 1\times D}$
S Score Tokens $\mathbb{R}^{M\times 1\times D}$
C CLIP Tokens $\mathbb{R}^{M\times 1\times D}$
SP Spatial Prompt Embeddings $\mathbb{R}^{M\times P\times D}$
Text Prompts
Refined CLIP Tokens
Refined Score Tokens
Refined Mask Tokens
Spatial Prompt Embeddings
Conditioned Point Embeddings
CLIP Embedding Head
Confidence Score Head
Mask Head
Encoder CLIP
CLIP Embeddings
Confidence Scores
Instance Masks
Auxiliary Masks
$\mathcal{L}_{\text{text}}$
$\mathcal{L}_{\text{score}}$
$\mathcal{L}_{\text{Dice}} + \mathcal{L}_{\text{focal}}$
$\mathcal{L}_{\text{aux}}$

(a) Prompt-Point Attention Module.

(b) Text Encoder & Prediction Heads.

Figure 6. **Detailed architecture of SNAP**. (a) The Prompt-Point Attention module refines both point and prompt embeddings within the Mask Decoder. (b) The refined embeddings are then fed into several lightweight prediction heads for mask, confidence score, and CLIP embedding predictions. For completeness, we also indicate the external CLIP Text Encoder for processing text prompts and the supervision signals associated with each branch.

7.3. Auto Prompt Generation

Given an input point cloud $\mathcal{P}$ with N points, let $\mathcal{F}$ denote the segmentation model and d the scene domain (outdoor / indoor / aerial). We define v_0 as the initial domain-specific voxel size, K_{max} as the maximum number of iterations, τ_s as the predicted confidence score threshold, and τ_{nms} as the NMS IoU threshold. The algorithm iteratively generates prompt points, segments objects of decreasing scales, and refines the results across iterations, as illustrated in **Fig. 7**. Finally, it outputs a set of masks $\mathcal{M}$, their corresponding text embeddings $\mathcal{T}$, and confidence scores $\mathcal{S}$. The complete procedure is summarized in **Alg. 1**.

8. Implementation Details

8.1. Details on Evaluation Metrics

IoU@k. Following conventions from [2, 3, 5], we evaluate using IoU@k, the average intersection over union (IoU) achieved with k clicks per object, averaged across all objects.

Average Precision. In our comparisons against the baselines for open-vocabulary segmentation, we use the Average Precision metric defined as follows -

$$\text{mAP} = \frac{1}{C \times 10} \sum_{c=1}^{C} \sum_{\tau=0.5}^{0.95} \text{AP}_{\tau}^{c} \quad \text{(step size = 0.05)} \qquad (6)$$

where C is the number of classes.

Panoptic Segmentation metrics. To assess panoptic quality, we utilize the Panoptic Segmentation metrics as defined in [40] and as used by SAL [7]. Specifically, PQ is Panoptic Quality, SQ is Segmentation Quality and RQ is Recognition Quality. TP, FP, FN represent True Positives, False Positives and False Negatives respectively. For class-aware segmentation, A prediction is counted as a TP if it has a IoU > 0.5 and the correct label. For class-agnostic segmentation, we assume class predictions are correct and TP is counted if IoU > 0.5.

Algorithm 1 Mask generation with iterative prompting algorithm.

Input: Point cloud $\mathcal{P} = \{p_i\}_{i=1}^{N}$, segmentation model $\mathcal{F}$, scene domain d, initial domain-specific voxel size v_0, maximum number of iterations K_{max}, confidence score threshold τ_s, NMS IoU threshold τ_{nms}

Output: Set of masks $\mathcal{M}$, text embeddings $\mathcal{T}$, confidence scores $\mathcal{S}$

```
$v_0 \leftarrow$ domain-specific voxel size
$\mathcal{C} \leftarrow \{0\}^N$                              ▷ Coverage mask
$\mathcal{M}, \mathcal{T}, \mathcal{S} \leftarrow \emptyset$
for $k = 0$ to $K_{\text{max}} - 1$ do
    $\mathcal{U} \leftarrow \{p_i : \mathcal{C}_i = 0\}$              ▷ Get uncovered points
    $v_k \leftarrow v_0 / 2^k$                            ▷ Halve voxel size
    $\mathcal{Q} \leftarrow \text{VoxelDownsample}(\mathcal{U}, v_k)$   ▷ Generate prompts
    $\mathcal{M}^k, \mathcal{T}^k, \mathcal{S}^k \leftarrow \mathcal{F}(\mathcal{P}, \mathcal{Q})$        ▷ Run model
    for $j = 1$ to $|\mathcal{M}^k|$ do
        if $\mathcal{S}_j^k \geq \tau_s$ then
            $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathcal{M}_j^k\}$
            $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathcal{T}_j^k\}$
            $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{S}_j^k\}$
            $\mathcal{C} \leftarrow \mathcal{C} \vee \mathcal{M}_j^k$             ▷ Update coverage
        end if
    end for
end for
$\mathcal{M}, \mathcal{T}, \mathcal{S} \leftarrow \text{NMS}(\mathcal{M}, \mathcal{T}, \mathcal{S}, \tau_{\text{nms}})$
return $\mathcal{M}, \mathcal{T}, \mathcal{S}$
```

Table 10. **Summary of Dataset Statistics.**

Training Datasets				
Dataset	Train	Val	Domain	Sensor Type
SemanticKITTI	19,130	4,071	Outdoor	HDL-64 LiDAR
nuScenes	28,130	6,019	Outdoor	32-beam LiDAR
PandaSet	2,000	400	Outdoor	Pandar64
ScanNet	1,201	312	Indoor	RGBD Camera
HM3D	1805	481	Indoor	RGBD camera
STPLS3D	3,395	500	Aerial	Photogrammetry
DALES	2,900	1,100	Aerial	Aerial LiDAR
Total	58,561	12,883		

Zero-Shot Validation Datasets			
Dataset	Val	Domain	Sensor Type
Waymo	5,976	Outdoor	Proprietary LiDAR
KITTI-360 SS	13,440	Outdoor	HDL-64 LiDAR
KITTI-360 Full	61	Outdoor	HDL-64 LiDAR
KITTI-360 Crops	3,421	Outdoor	HDL-64 LiDAR
Matterport3D	233	Indoor	RGBD Camera
ScanNet++	178	Indoor	RGBD Camera
S3DIS Crops	2,330	Indoor	RGBD Camera
S3DIS Full	68	Indoor	RGBD Camera
UrbanBIS	46	Aerial	Photogrammetry
Total	25,753		

$$\text{PQ} = \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{Recognition Quality (RQ)}} \times \underbrace{\frac{\sum_{(p,g)\in TP} \text{IoU}(p,g)}{|TP|}}_{\text{Segmentation Quality (SQ)}} \tag{7}$$

8.2. Dataset Details

We provide a summary of the dataset statistics in **Tab. 10**. Samples from each dataset illustrating the various domains are visualized in **Fig. 8**, **9** and **10**.

8.2.1 Indoor Datasets

ScanNetV2 [11] is a richly annotated dataset of 3D indoor scenes, covering a wide variety of scenes including offices, rooms, hotels etc. It provides semantic segmentation masks for 200 fine-grained classes, known as ScanNet200, and 20 coarser classes known as ScanNet20. We evaluate on both the benchmarks.

Habitat Matterport 3D [12] is a large scale annotated dataset for 3D indoor scenes which covers 216 3D spaces and 3100 rooms within these spaces. It provides instance annotation for 40 categories. After processing, it provides us with 1805 samples for training and 481 samples for validation.

ScanNet++ [17] comes with high-fidelity 3D mask annotations including smaller objects which are not well labeled in the ScanNet datasets. It includes high-resolution 3D scans captured at sub-millimeter precision and annotated comprehensively, covering objects of varying sizes.

Matterport3D [18] is a collection of 90 high-quality 3D reconstructions of indoor environments with instance annotations for 21 object categories. After processing, it provides us with 233 samples for validation.

S3DIS Full [19] is a collection of 6 large scale scenes covering 271 rooms. it provides annotations for 13 semantic classes. Following prior works [3, 4, 29] in instance segmentation, we use *Area_5* for evaluation which contains 68 samples for validation.

S3DIS Crops is proposed by AGILE3D [3] in their evaluation setting, cropping the validation samples from the original S3DIS dataset around each instance into 3m × 3m blocks. They provide the processed data on their github here. We call this dataset variant S3DIS Crops.

8.2.2 Outdoor Datasets

Outdoor datasets include two types of classes: *things*, which have instance labels, and *stuff*, which do not. This distinction can hinder the effectiveness of a promptable segmentation model. To address this, we use an off-the-shelf clustering algorithm, HDBSCAN [39], to provide us with pseudo instance labels for the *stuff* classes, enabling instance-wise promptable training on these categories. Details about HDBSCAN are provided in § 8.3.

SemanticKITTI [8] is derived from the KITTI Odometry [20] datasets. Each point in the dataset is densely labeled with one of $C = 19$ classes divided into *things* (with instance labels) and *stuff* (without instance labels) classes. We run HDBSCAN [39] to generate instance labels for the *stuff* classes.

nuScenes [9] is a comprehensive dataset that includes over 1000 diverse driving records, each lasting around 20 seconds. The LiDAR data from nuScenes is densely annotated with $C = 32$ classes, again divided into *things* and *stuff* classes. We run HDBSCAN [39] to generate instance labels for the *stuff* classes.

PandaSet [10] is an autonomous driving dataset featuring 103 driving sequences lasting about 8 seconds each. We only use a subset of 39 out of 103 scenes as the original dataset download links have expired. The dataset used for our training can be found at Kaggle. It provides semantic annotations for 37 classes divided into 28 *things* and 9 *stuff* classes. We run HDBSCAN [39] to generate instance labels for the *stuff* classes.

KITTI-360 Full [21] is a large-scale outdoor driving dataset which provides 360-degree annotations on point clouds, including bounding boxes, semantic, and instance annotations. The original dataset only provides labels for down-sampled superimposed point clouds. We call this original version KITTI-360 Full. This dataset provides annotations on 37 semantic classes in 19 object categories.

KITTI-360 Single Scan is derived from the KITTI-360 [21] Full dataset by following Interactive4D [5], where we applied a nearest-neighbor algorithm to propagate labels to individual points in individual scans. We use publicly available scripts for this purpose (Sanchez, 2021). We call this derived version KITTI-360 Single Scan. This also contains annotations for 37 semantic classes in 19 object categories. Since SNAP is trained using HDBSCAN-based instance labels for *stuff* classes, it would constitute a different evaluation setting if we evaluate on all classes. To keep evaluations fair, we only evaluate on the 11 *things* classes and compare against baselines.

KITTI-360 Crops is also derived from KITTI-360 [21] Full dataset. Specifically, to keep consistent with prior works like AGILE3D [3], which evaluated their indoor models on this variant, we also use the cropped version from their provided list of preprocessed scenes. This preprocessing includes dividing the original superimposed point clouds into smaller 3m × 3m chunks centered around the object instance.

Waymo [22] is a large-scale outdoor driving dataset which provides semantic labels across 23 classes but does not provide any instance annotations. However, Waymo does provide bounding boxes for 4 classes, including *vehicle, cyclist, sign*, and *pedestrian*. We use the combination of bounding boxes and semantic labels to generate instance labels for these 4 classes. After preprocessing on the entire validation set of Waymo, we get 5,976 samples for validation. Our preprocessed dataset will be released for reproducibility.

8.2.3 Aerial Datasets

STPLS3D [13] is a large-scale photogrammetry point cloud dataset covering approximately 16km^2 of urban and rural areas in Malaysia. Released in 2020, it contains over 2 billion labeled points across 25 scenes with annotations for 14 semantic classes. To keep computational demands tractable, we crop the point clouds to 50m × 50m blocks. The dataset is generated from aerial imagery using photogrammetric techniques, providing dense colored point clouds.

DALES [14] is a large-scale aerial LiDAR dataset covering 10km^2 of diverse landscapes, including urban, suburban, rural, and forested areas. It contains over 505 million points manually annotated with 8 semantic classes. Since DALES does not provide instance annotations, we again employ HDBSCAN [39] to generate instance labels for training and validation. We also crop the point clouds to 50m × 50m blocks. The dataset provides high-density aerial LiDAR data (50 points per m^2) captured at varying altitudes, making it particularly challenging due to large variations in point density and object scales.

UrbanBIS [24] is a dataset for large-scale 3D urban understanding, supporting practical urban-level semantic and building-level instance segmentation. UrbanBIS comprises six real urban scenes, with 2.5 billion points, covering a vast area of 10.78km^2 and 3,370 buildings, captured by 113,346 views of aerial photogrammetry. It provides annotations on 6 scenes, out of which we evaluate on the *Yingrenshi* test scenes. After cropping to 50m × 50m blocks, this provides us with 46 validation samples.

8.3. HDBSCAN Details

For the outdoor datasets used to train SNAP-C, the datasets include two types of classes: *things*, which have instance labels and *stuff* which do not have instance labels. From a promptable segmentation perspective, instance labels from *things* classes fit in directly. However, *stuff* includes classes such as vegetation, roads, buildings, *etc*., and is assigned a single label for all of them. The objects from these classes can be far away from each other and thus using one label directly is counterproductive in training a promptable segmentation model. To solve this issue, we propose to preprocess the datasets with HDBSCAN [39]. Specifically, we first take all the points belonging to a *stuff* class, and apply clustering on it. This helps in making multiple clusters from single class labels, which can then be used for promptable segmentation training.

8.4. Training and Inference Details

All SNAP variants are trained for 100 epochs using 8×NVIDIA A6000 GPUs. We train with a batch size of 1, where each batch corresponds to a single point cloud, from which 32 objects are randomly sampled for supervision. During training, we set the maximum click budget to 10. In each iteration, the number of clicks is randomly sampled between 1 and 10, ensuring that the model is consistently exposed to varying levels of user interaction. We use mixed-precision training to speed up both the training and evaluation process. We employ a round-robin style multi-dataset dataloader that repeats smaller datasets multiple times to keep the sample count similar to large datasets. During training, this dataloader provides a point cloud from one of the datasets at each training iteration, with each batch containing samples from a single dataset. To ensure proper routing through the correct normalization layer, we follow [15] and attach a *domain* variable to each point cloud. During training, this approach allows the network to route the data to the correct normalization layer. For interactive inference, this functionality translates into a simple domain-type checkbox selection, making it highly user-friendly.

Table 11. **Backbone Ablation on the ScanNet dataset.** Note that memory and time statistics are reported for 1-Click experiments.

Backbone	IoU@k ↑			Memory @1 click	Time @1 click
	@1	@5	@10		
AGILE3D [3]	63.3	79.9	83.7	1.2 GB	203 ms
Minkowski [41]	68.4	82.2	83.4	1.8 GB	213 ms
PTv3 [33]	68.6	82.1	84.6	1.3 GB	197 ms

Table 12. **Inference Click Strategy Ablations.** We evaluate different click strategies on the ScanNet20 dataset. Random Sampling represents all click points sampled randomly on the target object. Iterative sampling represents additional click points sampled in the unsegmented region from previous click mask.

Strategy	IoU@k ↑		Time (ms)
Random Sampling	@1	68.6	170
	@5	79.4	178
	@10	80.5	185
Iterative Sampling	@1	68.6	170
	@5	82.3	190
	@10	85.5	211

9. Additional Ablations

9.1. Effect of Backbone Architecture

We use the PTv3 [33] backbone for feature extraction, but a natural question to ask is, "How is the model performance affected if we use a different backbone?" To answer this, we compare PTv3 [33] with the Minkowski Res16UNet34C[41] backbone, which has been employed by [2, 3, 5]. The comparison, conducted on the ScanNet dataset, is summarized in **Tab. 11**. We observe consistent improvement in both PTv3 and Minkowski Engine backbones against AGILE3D [3], showing that our approach is equally applicable across both recent transformer-based as well as the common sparse-convolution-based backbones. To compute the memory and timing requirements, we use a random uniform point cloud with 100,000 points on all methods.

9.2. Effect of Click Strategy

We evaluate two click strategies during inference: (1) Random-sampling and (2) Iterative Refinement. The results are shown in **Tab. 12**, with timing measurements obtained by running the evaluation on a single NVIDIA RTX 3090 GPU. While Iterative Refinement performs much better than random sampling, it also runs slower in comparison. The random sampling strategy is especially helpful for users when trying to segment objects in the scene, because users can give multiple clicks at the beginning (which is equivalent to random sampling) to get a high-quality mask and later use refinement clicks to further improve the mask quality. In SNAP, we provide the flexibility to use both approaches during inference. To compute inference time, we used a point cloud from the ScanNet dataset with about 80,000 points.

9.3. Cross-Domain Input Ablation

To determine the effect of passing the wrong *domain input* when running zero-shot evaluations, we evaluate different domain settings of SNAP-C on 3 in-distribution and 6 zero-shot datasets. As demonstrated in **Tab. 13**, the correct domain input is crucial for getting good performance from the model on both in-distribution and zero-shot datasets. Moreover, while using the outdoor domain on indoor scenes completely disrupts performance, using the outdoor domain on aerial scenes still yields reasonable segmentation results, and vice versa. This is expected because aerial LiDAR captures are often collected over outdoor environments, which introduces partial similarities between aerial and outdoor domains while still retaining distinct characteristics.

10. Additional Quantitative Results

10.1. Timing and Memory Consumption Comparison

We compare the computational efficiency of SNAP with other interactive segmentation methods in **Tab. 14**, reporting inference time and memory consumption on an RTX 3090 GPU for single-object segmentation with 1 click. The results indicate that SNAP maintains competitive efficiency across both time and memory. For this test, we use the same uniform random point cloud with 100,000 points for all methods.

10.2. Class-Agnostic Interactive Segmentation against Non-interactive Fully-Supervised Methods

We evaluate our interactive model variants against state-of-the-art *non-interactive baselines* on both in-distribution and zero-shot datasets as a sanity check for their effectiveness. Since SNAP variants benefit from click supervision on all objects in the scene, they are expected to outperform non-interactive instance segmentation methods. As shown in **Tab. 15**, all SNAP variants achieve substantial gains over the current SOTA method EASE[29] on the ScanNet200 [11] benchmark, surpassing it by a 19.3 point margin with a single click and further improving as the number of clicks increases (5 and 10). This large advantage comes from the fact that predicting 200 categories, especially the long-tailed proportion, is inherently difficult for non-interactive methods, whereas SNAP benefits from click guidance that helps disambiguate object boundaries, leading to a much stronger performance. On the aerial STPLS3D [13] dataset with 14 semantic classes,

Table 13. **Effect of Cross-Domain Selection in Domain Normalization.** Evaluation results of SNAP-C when applying different domain types (Indoor, Outdoor, Aerial) in Domain Norm. The domain used for normalization is indicated in *teal* beneath each result.

Model	IoU @ k															
	In-Distribution						Zero-Shot									
	SemanticKITTI		ScanNet20		STPLS3D		Matterport3D		S3DIS Full		KITTI-360 Full		Waymo		UrbanBIS	
	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5
SNAP - C	**71.5**	**86.0**	19.2	52.3	41.4	66.9	17.4	45.6	15.7	51.3	**23.1**	**48.1**	**69.8**	**86.6**	55.2	80.1
Norm used	*Outdoor*		*Outdoor*		*Outdoor*		*Outdoor*		*Outdoor*		*Outdoor*		*Outdoor*		*Outdoor*	
SNAP - C	7.2	19.1	**67.7**	**82.3**	4.9	6.6	**52.6**	**75.2**	**53.6**	**77.6**	2.4	4.1	1.2	7.8	19.9	27.9
Norm used	*Indoor*		*Indoor*		*Indoor*		*Indoor*		*Indoor*		*Indoor*		*Indoor*		*Indoor*	
SNAP - C	27.1	57.9	11	22.1	**67.8**	**80.4**	12.6	22.3	8.2	25.6	6.8	28.3	25.1	60.4	**71.6**	**90.2**
Norm used	*Aerial*		*Aerial*		*Aerial*		*Aerial*		*Aerial*		*Aerial*		*Aerial*		*Aerial*	

Table 14. **Model Efficiency Comparison Results.** We compare the timing and memory consumption on an RTX 3090 GPU when performing single-object segmentation with 1 click.

Method	Model Size (M)	Memory (GB)	Inference Time (ms)
AGILE3D	39.3	1.20	203
Point-SAM	311.0	3.70	287
Interactive4D	39.3	1.05	200
SNAP	49.6	1.27	197

SNAP-C@1 Click shows slightly better performance than the current SOTA methods, and as expected, this continues to improve with additional clicks. On the outdoor SemanticKITTI [8] dataset, SNAP variants show very strong performance, significantly outperforming the SOTA Mask4Former [43] with 1-Click.

When evaluating zero-shot on unseen datasets like ScanNet++ [17] and Matterport3D [18], SNAP outperforms these method by 8.6 points on ScanNet++ and 14.5 points on Matterport3D. Notably, both the baseline methods LaSSM [44] and ODIN [45] were trained on ScanNet++ and Matterport3D datasets respectively. Further on the aerial UrbanBIS [24] dataset, SNAP again outperforms the B-Seg [24] baseline which was trained on the dataset. With additional clicks, this performance continues to improve.

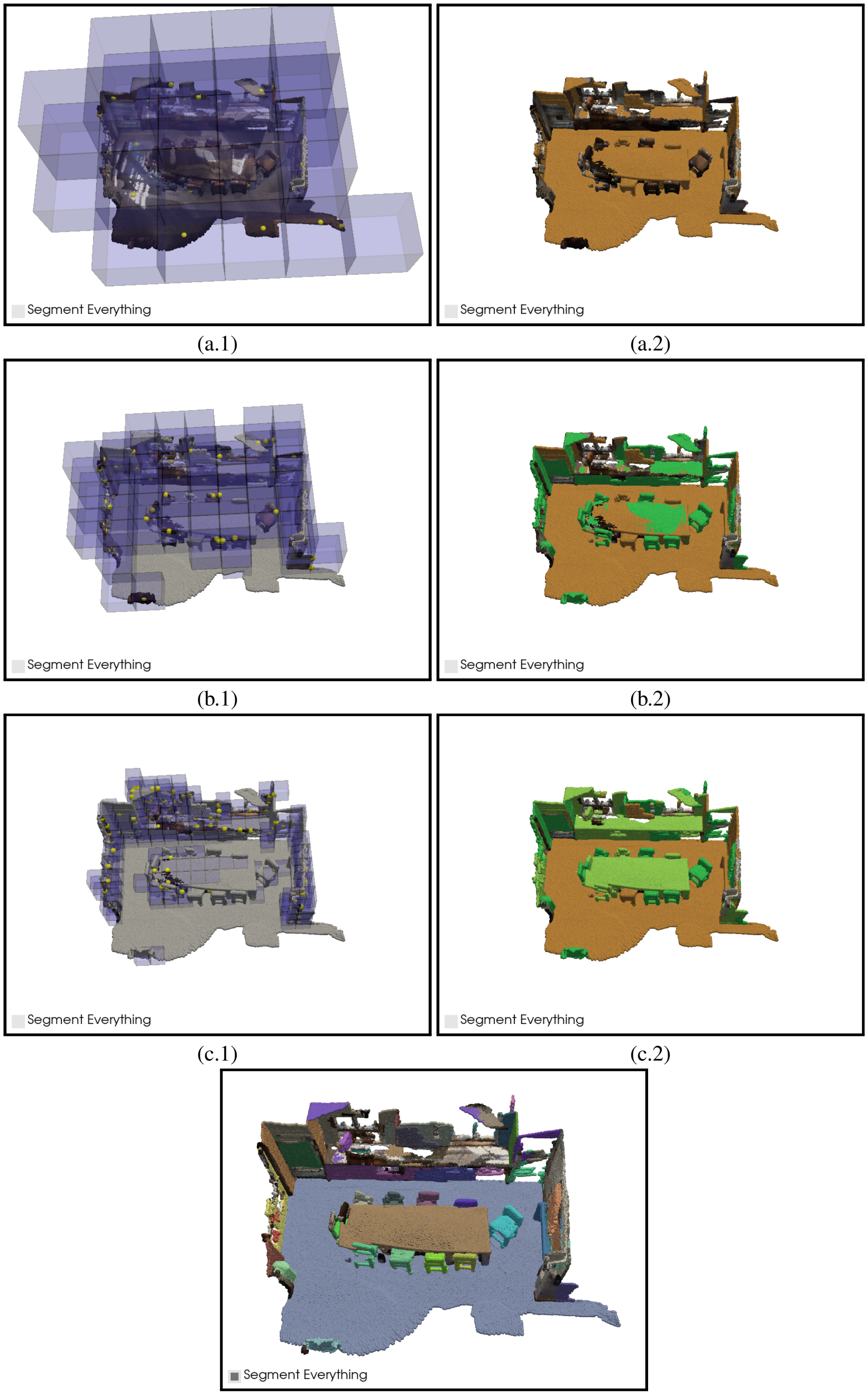
10.3. Interactive segmentation results with all model variants

Tab. 16 presents results for all SNAP variants on in-distribution datasets. For datasets lacking established baselines, we compare against zero-shot results from single-dataset models and in-distribution results from single-domain models. SNAP-C achieves the best 1-click performance on 4/7 datasets, best 3-click results on 6/7 datasets, and optimal performance across all datasets for higher click counts, demonstrating effective performance with a unified model.

Tab. 17 evaluates all SNAP variants on unseen datasets. SNAP-C outperforms baselines on 6/9 datasets for 1-click performance and maintains strong performance across different click counts (7/9 for 3-click, 6/9 for 5-click, 7/9 for 7-click, and 7/9 for 10-click). This demonstrates robust generalization across diverse domains. Notably, while SNAP-C may not achieve state-of-the-art performance on every individual dataset, it is the only method that operates across all domains with a single set of weights, unlike approaches such as AGILE3D [3] that require separate models for different scene types.

11. Additional Qualitative Results

We provide additional qualitative results, showcasing the performance of our SNAP model across different tasks. **Fig. 11** demonstrates the model's capability in open-vocabulary scene understanding on the ScanNet++ dataset by using arbitrary queries involving object categories that are not present during training. **Fig. 12-20** present point-based segmentation results on the ScanNet, SemanticKITTI, and STPLS3D datasets, respectively. For each domain, we compare the ground truth masks with our model's outputs under 1-click, 5-click, and 10-click interaction settings, reporting the corresponding mean IoU values. These results demonstrate the effectiveness of our approach in diverse environments, emphasizing the flexibility and robustness of our method across diverse segmentation challenges.



(a.1) (a.2)

(b.1) (b.2)

(c.1) (c.2)

(d)

Figure 7. **Visualization of the Iterative Prompting algorithm. (a.1):** Generate prompt points with a large voxel size to segment out large objects in the scene, **(a.2):** All the points segmented after first iteration (Yellow color). **(b.1):** Reduce the voxel size and generate prompt points on the unsegmented points. **(b.2):** All points segmented after Iteration 1 (yellow) and 2(dark green). **(c.1):** Reduce voxel size again and repeat. **(c.2):** All points segmented after Iteration 1 (yellow), 2(dark green) and 3 (light green). **(d):** Final instance masks after Non Maximum Suppression.

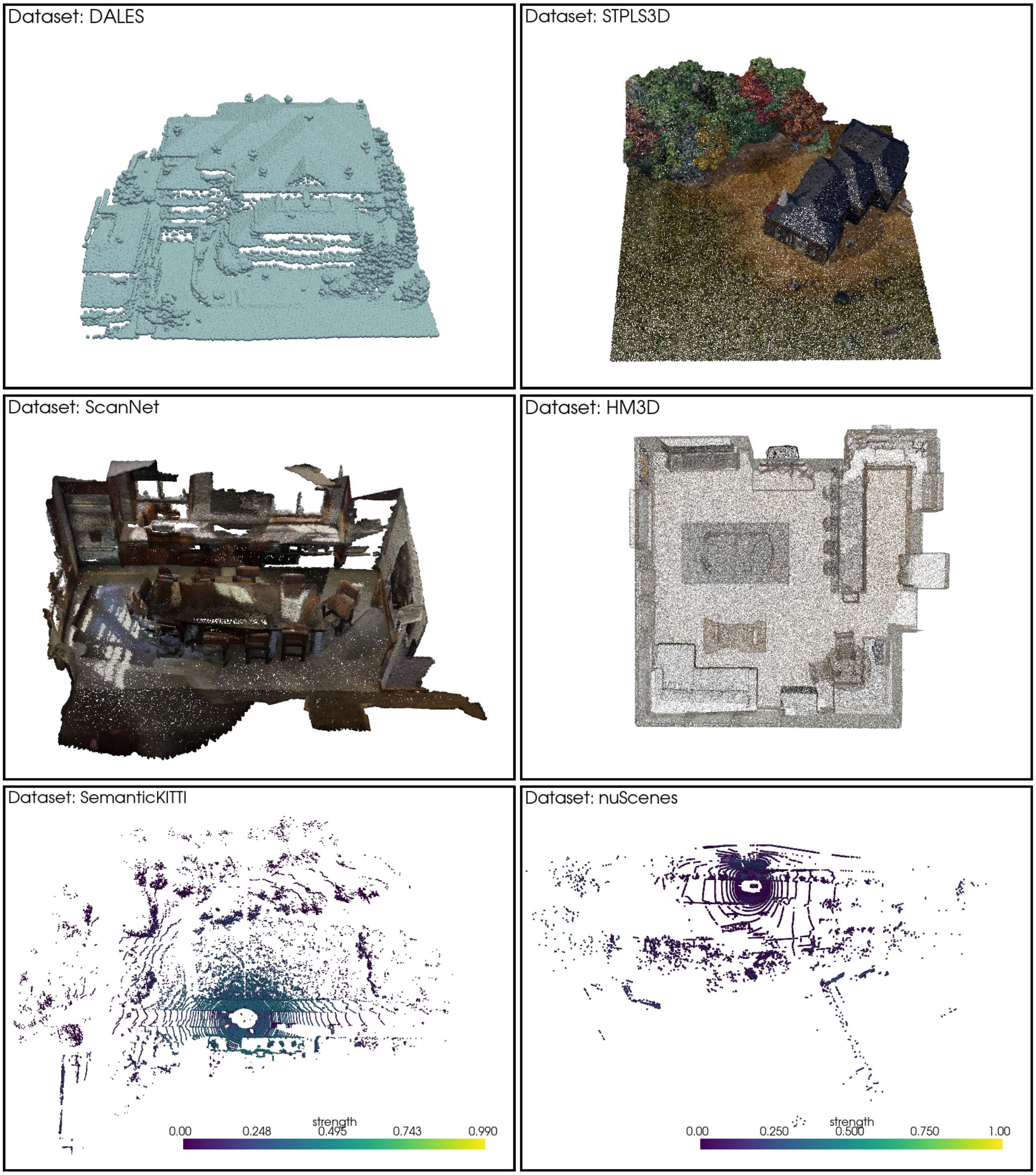


Figure 8. **Samples from Training Datasets.** Here we present samples taken from the training datasets to showcase the difference in scale, point density and scene types. Note that the dataset name is displayed in each figure. From the top - DALES and STPLS3D are aerial datasets, ScanNet and HM3D are indoor scene datasets and SemanticKITTI and nuScenes are outdoor scene datasets. HM3D provides full room scenes, point size has been reduced for better understanding of the scene.

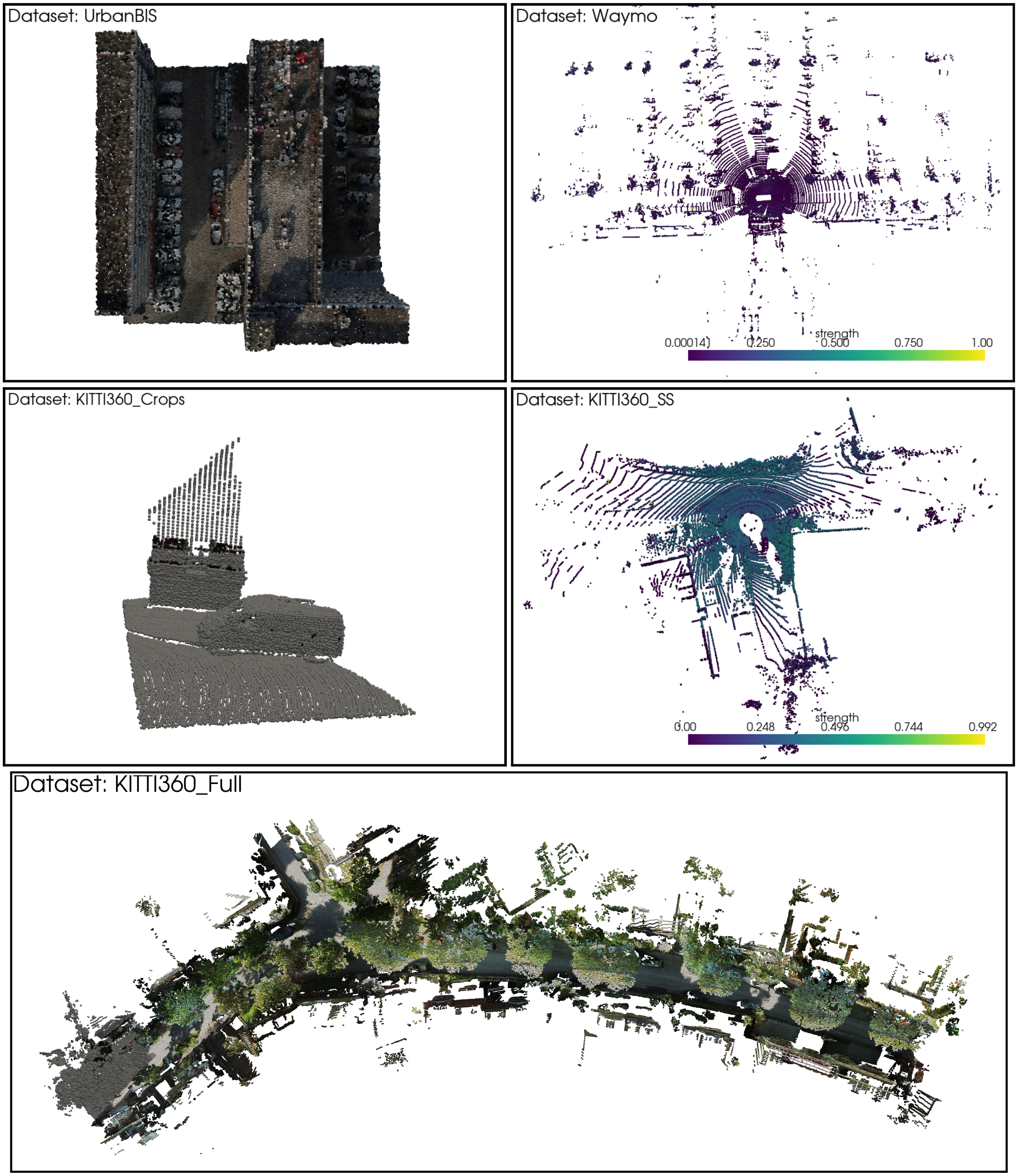


Figure 9. **Samples from Validation Datasets.** We evaluate SNAP on a variety of datasets. From the top - UrbanBIS is an Aerial scene; Waymo, KITTI-360 Crops, KITTI-360 Single Scan, and KITTI-360 Full are outdoor scene datasets. Note the difference in the variants of KITTI-360. KITTI-360 Crops particularly represents small-scale dense scenes generally found in indoor point clouds, while KITTI-360 Single Scan shows a traditional point cloud collected with a LiDAR sensor, and KITTI-360 Full shows an aggregated point cloud map built by combining multiple LiDAR scans.

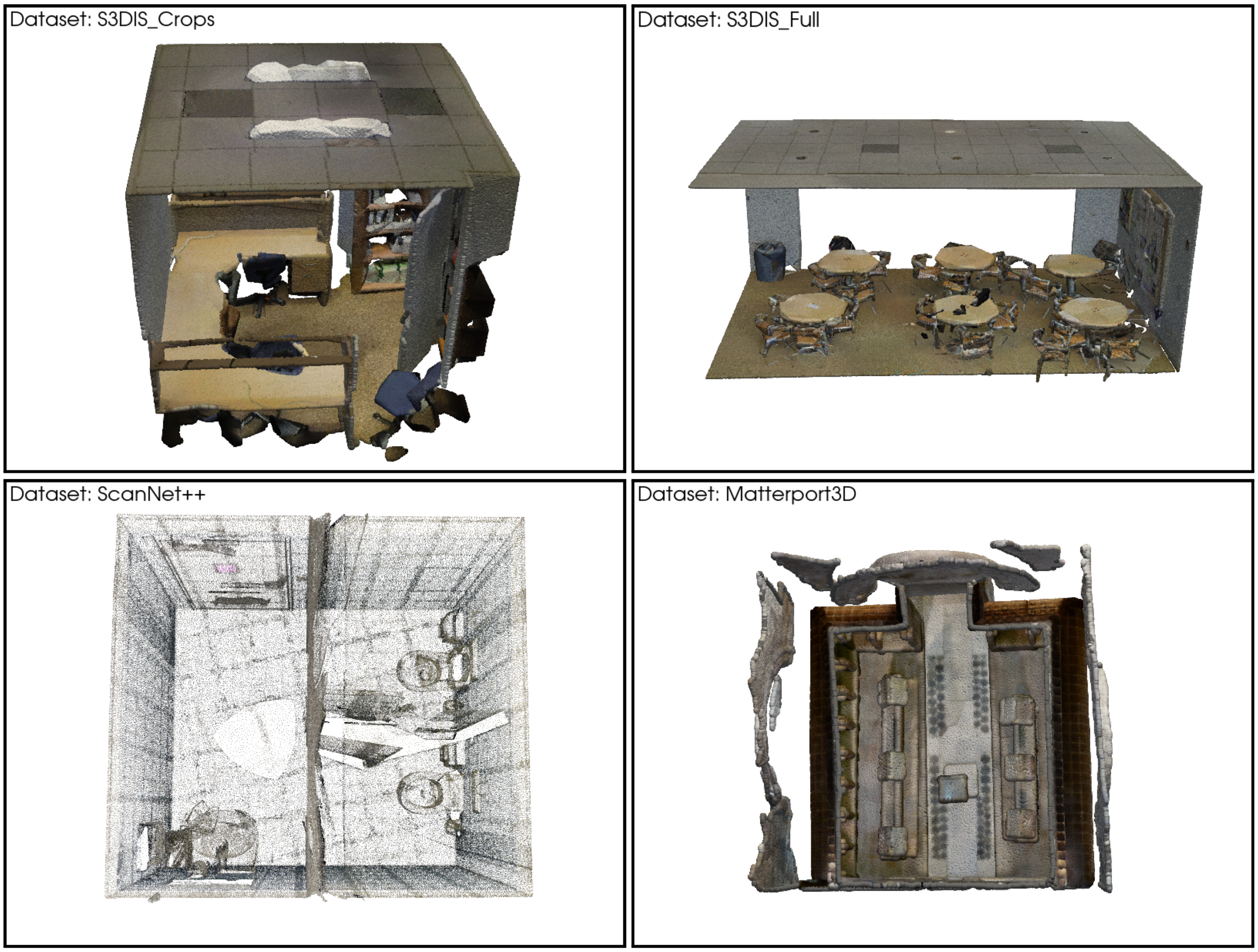


Figure 10. **Samples from Validation Datasets.** Here we show the examples from Indoor datasets used for validation. Note that S3DIS Crops is a cropped version of the full S3DIS point clouds, therefore some objects appear truncated. ScanNet++ provides full room scenes, point size has been reduced for better understanding of the scene.

Table 15. **Class Agnostic Instance Segmentation Comparison against Non-Interactive Fully-Supervised Methods**. We compare SNAP against state-of-the-art baselines for in-distribution and zero-shot datasets on the instance segmentation task in a class-agnostic fashion.

In-distribution Evaluation

ScanNet200 Validation Set

Method	mAP			mAP50			mAP25		
EASE [29]	29.9			38.8			44.7		
	@1	@5	@10	@1	@5	@10	@1	@5	@10
SNAP - SN200	47.9	68.1	73	69.7	89.6	92.4	84.2	97.3	98.8
SNAP - Indoor	45.2	66.7	73.3	69.3	90.5	95.3	84.8	98.8	99.7
SNAP - C	**49.2**	**69.8**	**77.5**	**73.2**	**91.6**	**95.9**	**87.7**	**99.1**	**99.8**

STPLS3D Validation Set

Method	mAP			mAP50			mAP25		
Mask3D [42]	57.3			74.3			81.6		
	@1	@5	@10	@1	@5	@10	@1	@5	@10
SNAP - Aerial	56.2	72.9	80.7	74.4	88.9	94	86.5	97.1	98.6
SNAP - C	**58.3**	**75.7**	**84.4**	**76.7**	**91.1**	**95.3**	**88.8**	**98.0**	**99.1**

Semantic KITTI Validation Set

Method	PQ			SQ			RQ		
Mask4Former [43]	61.7			81			71.4		
	@1	@5	@10	@1	@5	@10	@1	@5	@10
SNAP - KITTI	68.6	83.4	87.6	80.9	84.9	88.9	82.7	91.8	97.5
SNAP - Outdoor	69.9	85.1	90.1	82.3	87.9	91.3	84.1	96.3	98.3
SNAP - C	**71.1**	**86.5**	**90.7**	**82.7**	**88.7**	**91.7**	**84.8**	**97.4**	**98.4**

Zero-Shot Evaluation

ScanNet++ Validation Set

Method	mAP			mAP50			mAP25		
LaSSM [44]	29.1			43.5			51.6		
	@1	@5	@10	@1	@5	@10	@1	@5	@10
SNAP - SN200	32.9	52.1	58.1	49.1	71.9	77.3	62.9	86.1	89.6
SNAP - Indoor	35.9	59.8	66.5	**55.5**	**84.9**	89.4	**73.3**	**97.3**	**98.6**
SNAP - C	**37.7**	**60.4**	**70.1**	55.4	83.0	**90.9**	73.2	94.6	98.1

Matterport3D Validation Set

Method	mAP			mAP50			mAP25		
ODIN [45]	24.7			–			63.8		
	@1	@5	@10	@1	@5	@10	@1	@5	@10
SNAP - SN200	**42.7**	62.3	68.8	**64.4**	85.6	90.9	**77.5**	95.9	97.1
SNAP - Indoor	36.5	63	70.7	59.2	**90.3**	93.3	74.2	**98.1**	**98.8**
SNAP - C	39.2	**64.6**	**74.3**	59.4	88.7	**94.5**	77.3	96.9	98.7

UrbanBIS Validation Set

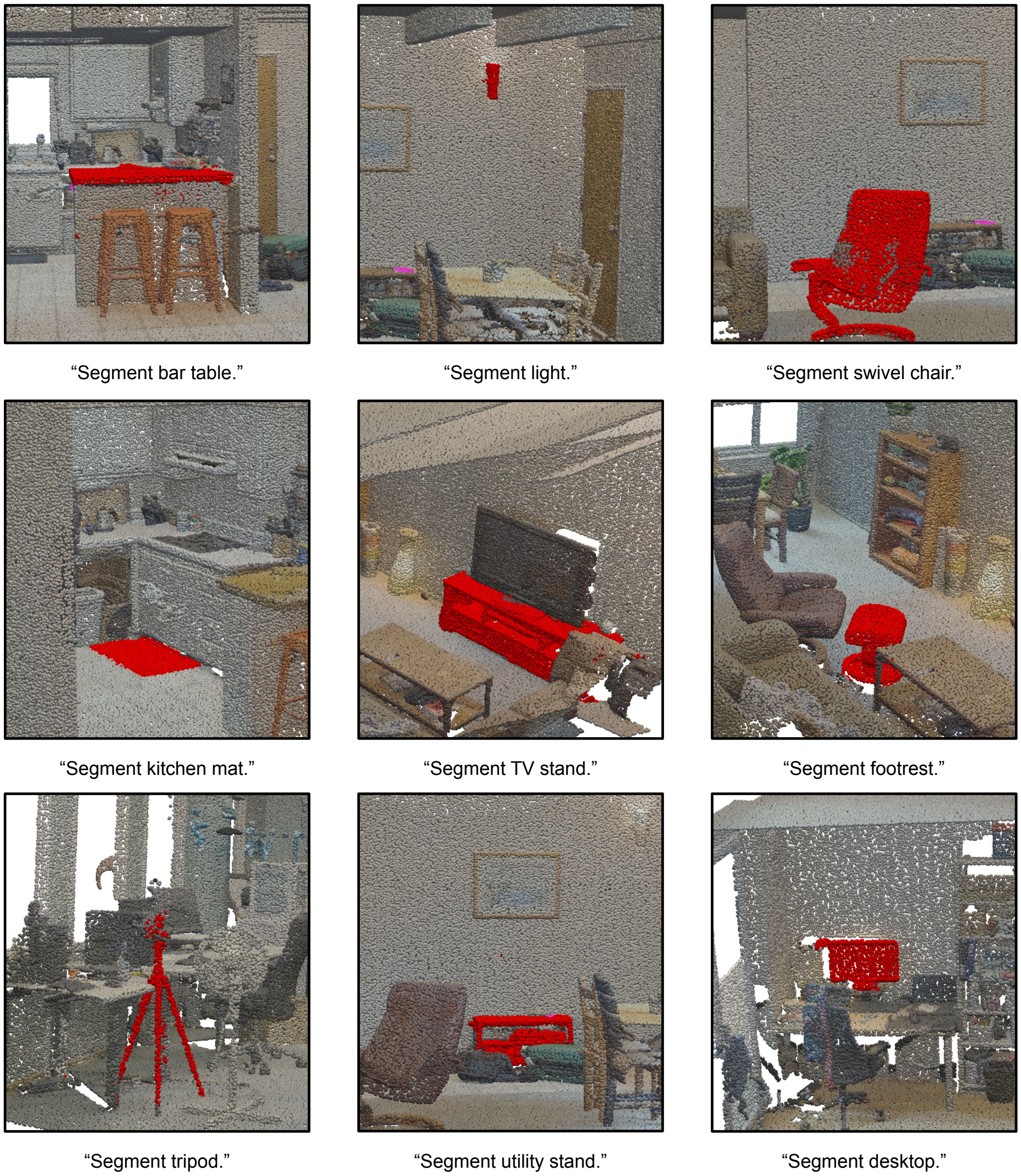
Method	mAP			mAP50			mAP25		
B-Seg [24]	62.1			70			73.9		
	@1	@5	@10	@1	@5	@10	@1	@5	@10
SNAP - Aerial	**62.9**	84.2	91.1	**85.6**	95.5	98.2	**96.4**	99.1	98.2
SNAP - C	62.2	**89.1**	**94.9**	84.1	**100**	**100**	94.6	**100**	**100**

Table 16. **In-distribution Interactive Point Cloud Segmentation Results.** * indicates models not trained on the evaluation dataset and † denotes that the methods are evaluated by us.

Domain	**Dataset**	**Method**	**IoU@k**				
			@1	**@3**	**@5**	**@7**	**@10**
Outdoor	SemanticKITTI	AGILE3D [3]	53.1	70	76.7	-	83
		Interactive4D [5]	67.5	78.3	83.4	-	88.2
		SNAP-KITTI	68.1	80.1	84.5	87.5	88.7
		SNAP-Outdoor	71.3	**81.9**	85.7	87.7	89.3
		SNAP-C	**71.5**	**81.9**	**86**	**88.1**	**90**
	nuScenes	AGILE3D* [3]	32.4	47.1	56.4	-	68.4
		Interactive4D* [5]	45.5	57.2	64.6	-	74.3
		SNAP-KITTI*	50.2	64.3	71.3	74.6	76.9
		SNAP-Outdoor	**72.4**	83.1	**88.1**	90.2	91.2
		SNAP-C	72.2	**83.3**	**88.1**	**90.3**	**92.2**
	Pandaset	SNAP-KITTI	17	29.7	34.2	35.8	36.7
		SNAP-Outdoor	**60.5**	**74.8**	80.1	82.1	84.3
		SNAP-C	56.3	74.6	**80.2**	**82.6**	**84.4**
Indoor	ScanNet20	InterObject3D [2]	40.8	63.9	72.4	-	79.9
		AGILE3D [3]	63.3	75.4	79.9	-	83.7
		Point-SAM† [4]	52.7	75.9	80.6	82.9	83.3
		SNAP-SN	**68.6**	78.4	82.1	83.4	84.6
		SNAP-Indoor	66	77.6	81.3	83	84
		SNAP-C	67.7	**78.5**	**82.3**	**84.1**	**85.5**
	HM3D	SNAP-SN	38.7	52.4	58.6	61.4	63.3
		SNAP-Indoor	47.1	65.2	71.2	74.0	75.9
		SNAP-C	**50**	**66.7**	**72.9**	**76.1**	**78.7**
Aerial	STPLS3D	SNAP-Aerial	65.8	74.5	79.1	81.6	83.6
		SNAP-C	**67.8**	**75.5**	**80.4**	**83.3**	**85.8**
	DALES	SNAP-Aerial	60.7	72.5	76.8	78.7	80
		SNAP-C	**61.6**	**74**	**78.2**	**80.4**	**82.3**

Table 17. **Zero-Shot Interactive Point Cloud Segmentation Results.** † denotes that the methods are evaluated by us.

Domain	**Dataset**	**Method**	**IoU@k**				
			@1	**@3**	**@5**	**@7**	**@10**
Outdoor	Waymo	Interactive4D	7.2	7.3	7.5	7.7	7.9
		Point-SAM	12.8	43	53.1	57.4	60.2
		SNAP-KITTI	48.2	61.8	66.3	68.4	70
		SNAP-Outdoor	68.5	81.8	86	87.4	88.3
		SNAP-C	**69.8**	**82.3**	**86.6**	**88.2**	**89.3**
	KITTI 360 Full	Point-SAM	6.8	22.7	28.1	30.7	32.5
		SNAP-KITTI	6.7	25.9	29.7	31.2	32.4
		SNAP-Outdoor	18.3	35.2	44.6	47.3	50.2
		SNAP-C	**23.1**	**40.1**	**48.1**	**51.7**	**54.2**
	KITTI 360 Single Scan	AGILE3D	36.3	47.3	53.5	-	63.3
		Interactive4D	47.7	59.4	64.1	-	70
		SNAP-KITTI	54.4	60.9	63.9	65.5	66.8
		SNAP-Outdoor	59.8	63.3	65.9	67.5	69.1
		SNAP-C	**60.4**	**64.6**	**67.7**	**70.1**	**72.6**
	KITTI 360 Crops	AGILE3D	34.8	42.7	44.4	45.8	49.6
		Point-SAM	49.4	74.4	**81.7**	**84.3**	**85.8**
		SNAP-KITTI	56.1	68.8	72.8	74.3	75.3
		SNAP-Outdoor	56.9	70.6	75.6	78.1	80.3
		SNAP-SN	54.5	68.6	74.1	76.8	78.6
		SNAP-Indoor	54.9	70.2	76.6	79.3	80.4
		SNAP-C	**65.6**	**76.1**	80	82.1	83.6
Indoor	ScanNet++	Point-SAM†	28.6	56.3	62.9	65.5	67.2
		SNAP-SN	45.5	59.9	65.3	67.8	69.5
		SNAP-Indoor	51.5	**67.9**	**73.4**	75.9	77.6
		SNAP-C	**52**	67.3	73.2	**76.3**	**78.6**
	Matterport3D	Point-SAM†	41.1	67.2	73.7	76.2	77.9
		SNAP-SN	**53.4**	66.6	71.3	73.7	75.3
		SNAP-Indoor	49.9	68.4	74.2	76.4	78.3
		SNAP-C	52.6	**69.6**	**75.2**	**78.2**	**80.5**
	S3DIS Crops	AGILE3D	**58.7**	77.4	83.6	86.4	**88.5**
		Point-SAM	45.9	**77.6**	**84.6**	**86.9**	88.4
		SNAP-SN	55.8	68.7	74.1	77.2	79.4
		SNAP-Indoor	54.8	73.5	80.5	83.7	85.9
		SNAP-C	56.6	73.8	80.9	84.4	87
	S3DIS Full	Point-SAM†	35.6	68	76.3	78.9	80.6
		SNAP-SN	51.4	64.3	70	72.6	74.8
		SNAP-Indoor	51.9	70.1	76.9	79.9	81.9
		SNAP-C	**53.6**	**71.1**	**77.6**	**80.8**	**83.2**
Aerial	UrbanBIS	Point-SAM	39.3	79.1	89.4	92.7	94.3
		SNAP-Aerial	**74.2**	83.2	86.9	89.8	90.6
		SNAP-C	71.6	**86.2**	**90.2**	**92.8**	**94.7**



"Segment bar table."

"Segment light."

"Segment swivel chair."

"Segment kitchen mat."

"Segment TV stand."

"Segment footrest."

"Segment tripod."

"Segment utility stand."

"Segment desktop."

Figure 11. **Additional qualitative segmentation results of open-set scene understanding on the ScanNet++ Dataset.** Given a text prompt in the format of "*Segment* {*open-set vocabulary*}", our SNAP model finds the corresponding masks 🟥 in the scenes.

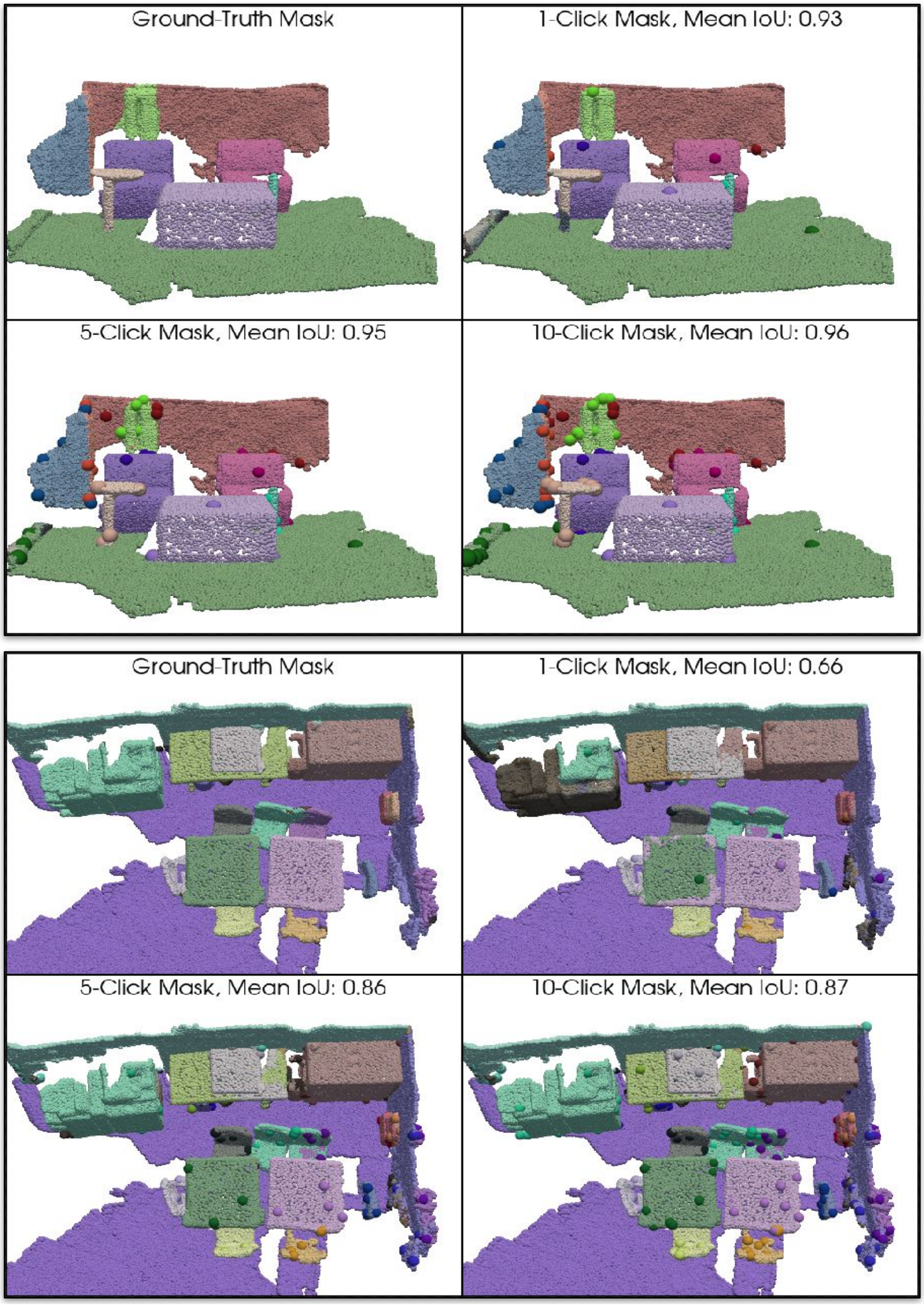

Ground-Truth Mask
1-Click Mask, Mean IoU: 0.93
5-Click Mask, Mean IoU: 0.95
10-Click Mask, Mean IoU: 0.96
Ground-Truth Mask
1-Click Mask, Mean IoU: 0.66
5-Click Mask, Mean IoU: 0.86
10-Click Mask, Mean IoU: 0.87

Figure 12. **Additional qualitative results for point-based segmentation on the ScanNet dataset.** For each block, we show the ground truth masks alongside our segmentation results for 1-click, 5-click, and 10-click interactions, including the corresponding mean IoU values. Points with the same color represent the same object, while clicks are highlighted using darker colors and larger spheres for better visibility.

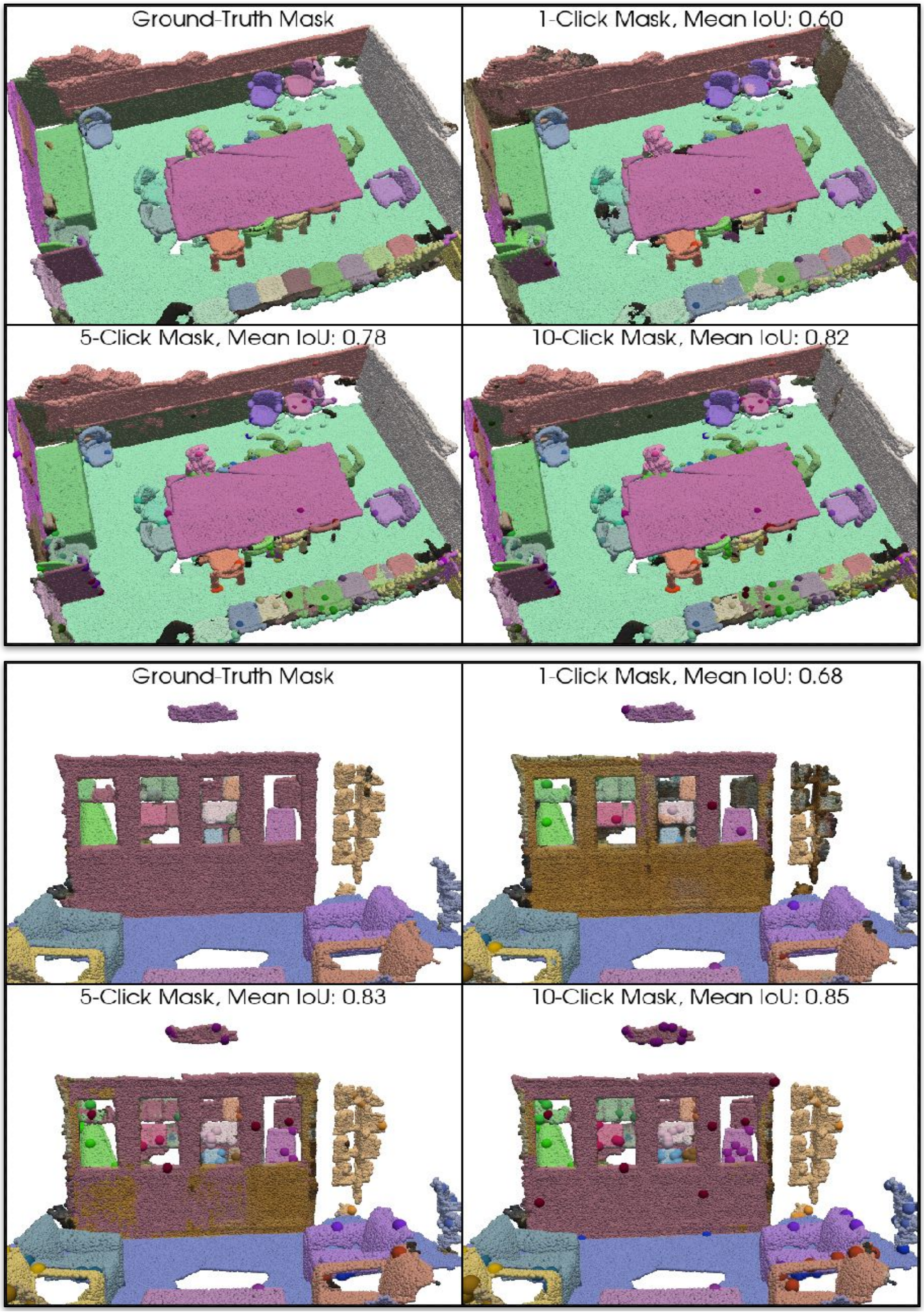

Ground-Truth Mask
1-Click Mask, Mean IoU: 0.60
5-Click Mask, Mean IoU: 0.78
10-Click Mask, Mean IoU: 0.82
Ground-Truth Mask
1-Click Mask, Mean IoU: 0.68
5-Click Mask, Mean IoU: 0.83
10-Click Mask, Mean IoU: 0.85

Figure 13. **Additional qualitative results for point-based segmentation on the ScanNet dataset.** For each block, we show the ground truth masks alongside our segmentation results for 1-click, 5-click, and 10-click interactions, including the corresponding mean IoU values. Points with the same color represent the same object, while clicks are highlighted using darker colors and larger spheres for better visibility.

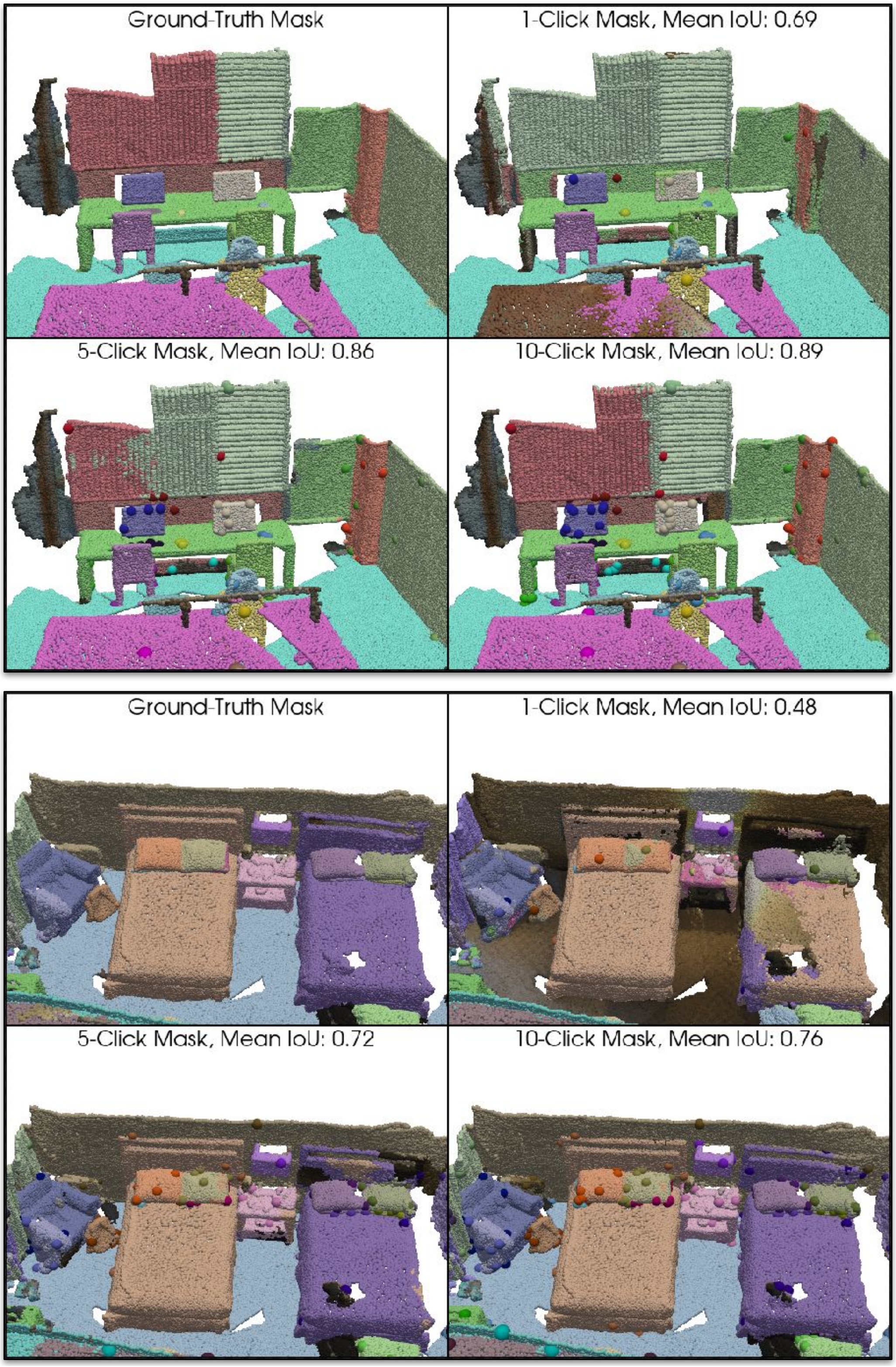

Ground-Truth Mask
1-Click Mask, Mean IoU: 0.69
5-Click Mask, Mean IoU: 0.86
10-Click Mask, Mean IoU: 0.89
Ground-Truth Mask
1-Click Mask, Mean IoU: 0.48
5-Click Mask, Mean IoU: 0.72
10-Click Mask, Mean IoU: 0.76

Figure 14. **Additional qualitative results for point-based segmentation on the ScanNet dataset.** For each block, we show the ground truth masks alongside our segmentation results for 1-click, 5-click, and 10-click interactions, including the corresponding mean IoU values. Points with the same color represent the same object, while clicks are highlighted using darker colors and larger spheres for better visibility.

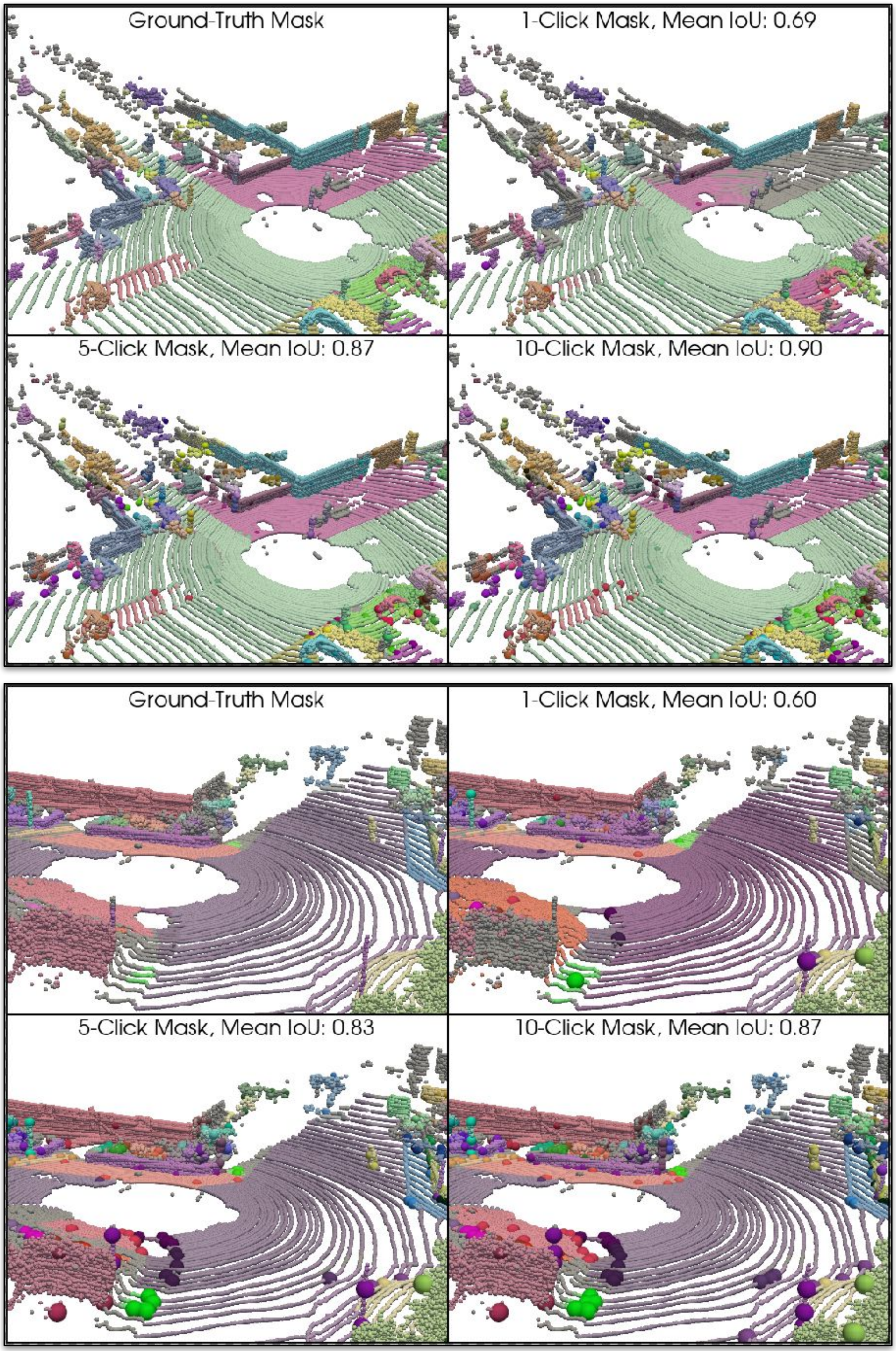

Ground-Truth Mask
1-Click Mask, Mean IoU: 0.69
5-Click Mask, Mean IoU: 0.87
10-Click Mask, Mean IoU: 0.90
Ground-Truth Mask
1-Click Mask, Mean IoU: 0.60
5-Click Mask, Mean IoU: 0.83
10-Click Mask, Mean IoU: 0.87

Figure 15. **Additional qualitative results for point-based segmentation on the SemanticKITTI dataset.** For each block, we show the ground truth masks alongside our segmentation results for 1-click, 5-click, and 10-click interactions, including the corresponding mean IoU values. Points with the same color represent the same object, while clicks are highlighted using darker colors and larger spheres for better visibility.

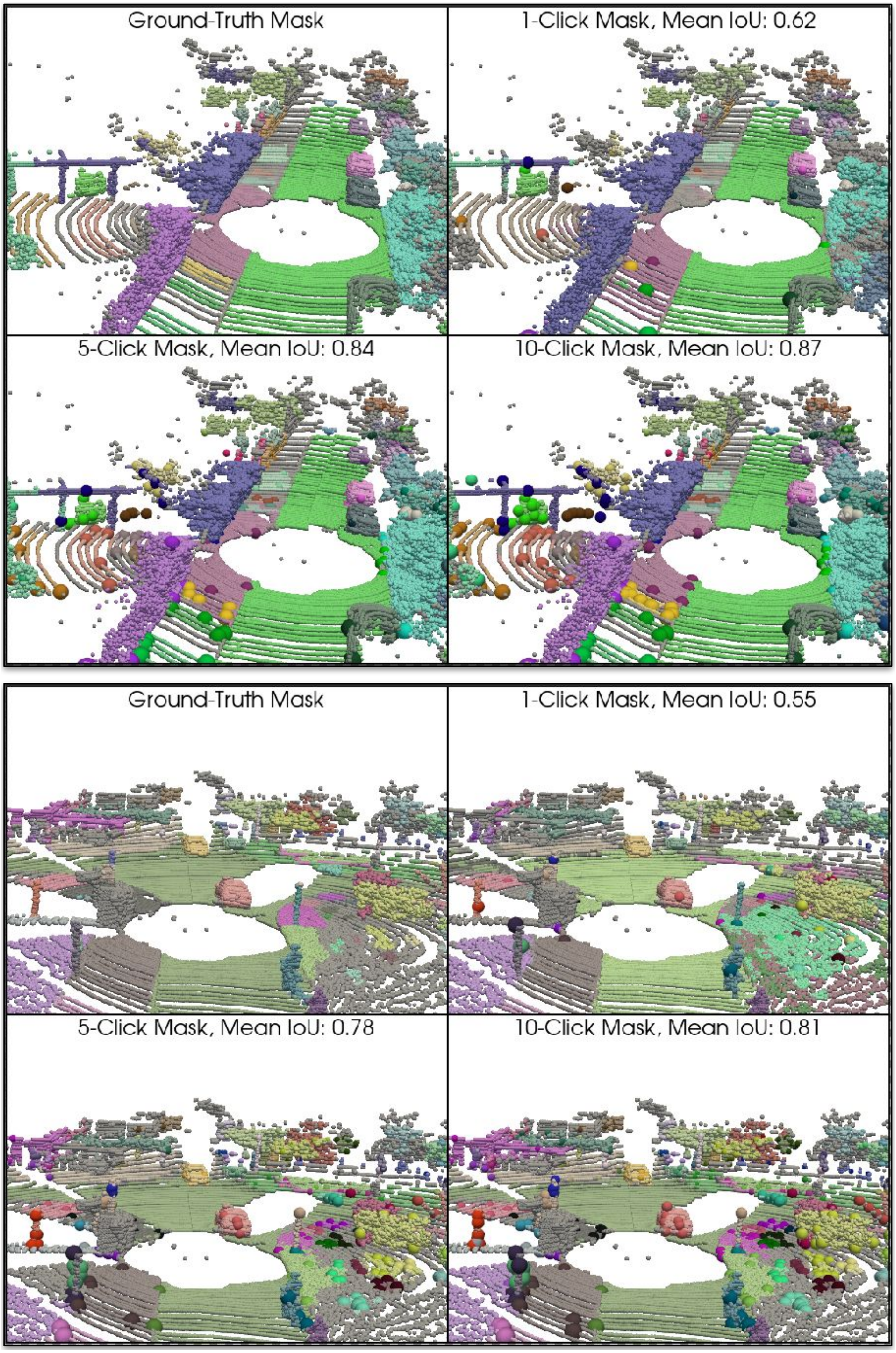

Ground-Truth Mask
1-Click Mask, Mean IoU: 0.62
5-Click Mask, Mean IoU: 0.84
10-Click Mask, Mean IoU: 0.87
Ground-Truth Mask
1-Click Mask, Mean IoU: 0.55
5-Click Mask, Mean IoU: 0.78
10-Click Mask, Mean IoU: 0.81

Figure 16. **Additional qualitative results for point-based segmentation on the SemanticKITTI dataset.** For each block, we show the ground truth masks alongside our segmentation results for 1-click, 5-click, and 10-click interactions, including the corresponding mean IoU values. Points with the same color represent the same object, while clicks are highlighted using darker colors and larger spheres for better visibility.

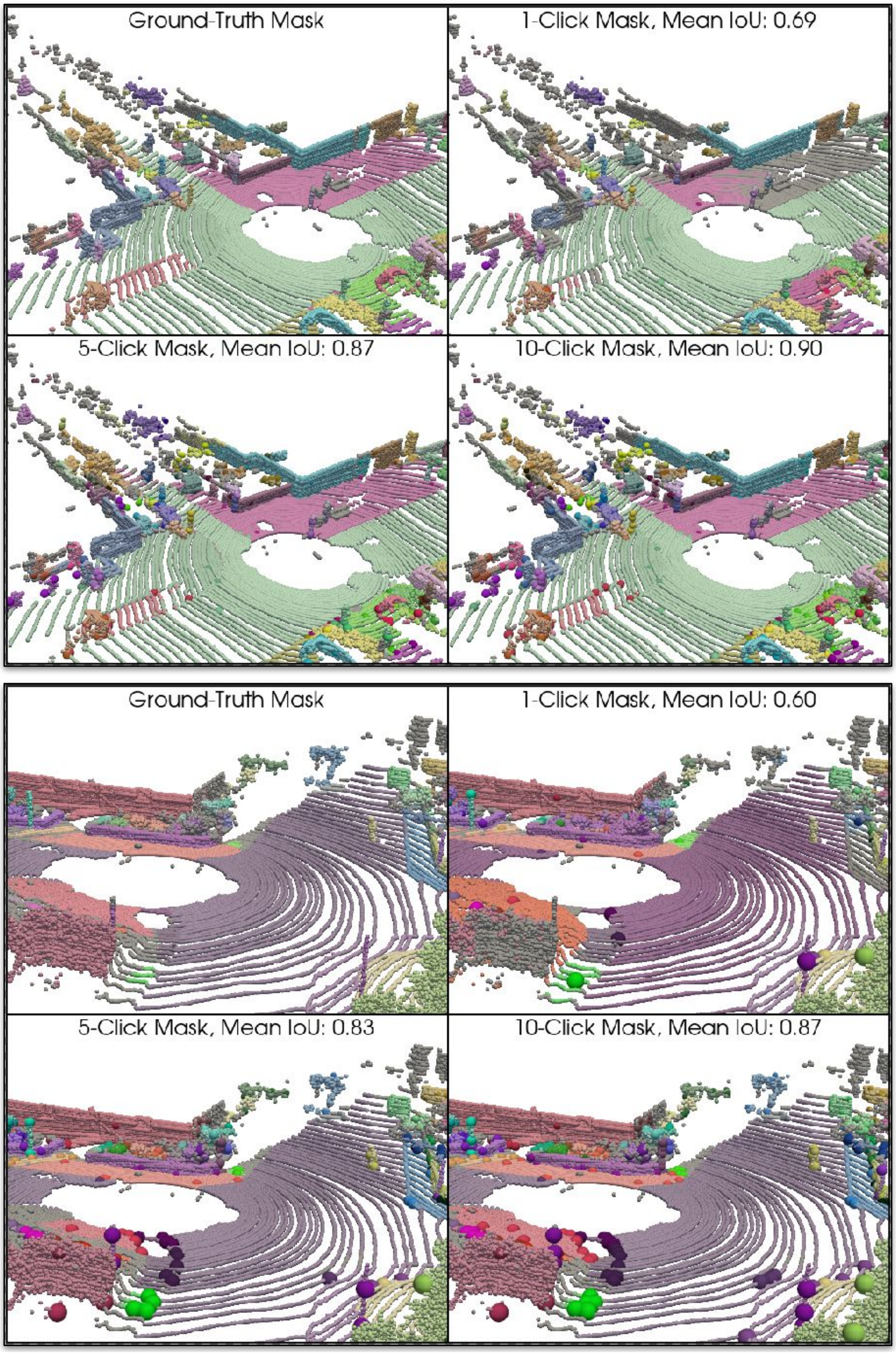

Ground-Truth Mask
1-Click Mask, Mean IoU: 0.69
5-Click Mask, Mean IoU: 0.87
10-Click Mask, Mean IoU: 0.90
Ground-Truth Mask
1-Click Mask, Mean IoU: 0.60
5-Click Mask, Mean IoU: 0.83
10-Click Mask, Mean IoU: 0.87

Figure 17. **Additional qualitative results for point-based segmentation on the SemanticKITTI dataset.** For each block, we show the ground truth masks alongside our segmentation results for 1-click, 5-click, and 10-click interactions, including the corresponding mean IoU values. Points with the same color represent the same object, while clicks are highlighted using darker colors and larger spheres for better visibility.

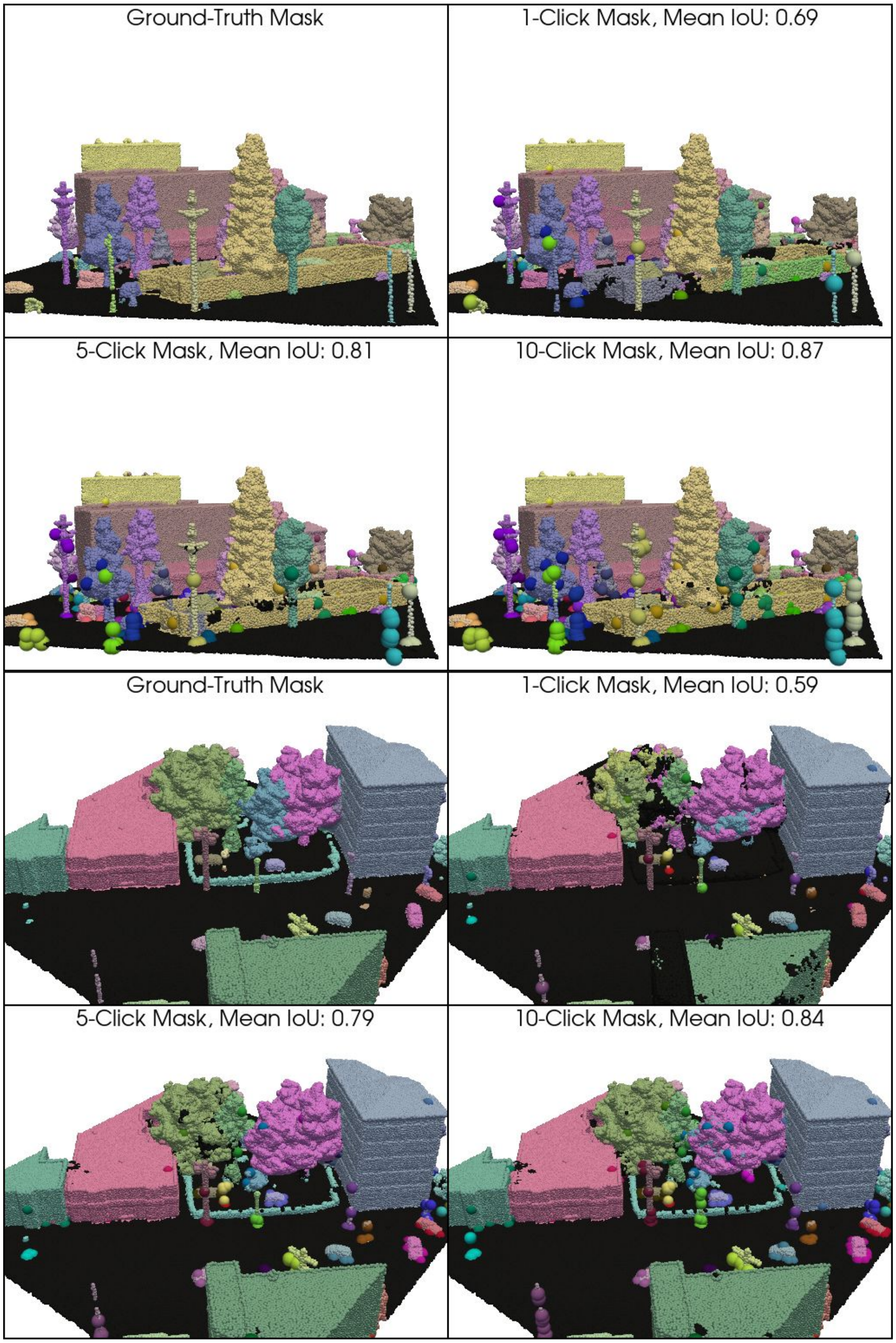

Ground-Truth Mask
1-Click Mask, Mean IoU: 0.69
5-Click Mask, Mean IoU: 0.81
10-Click Mask, Mean IoU: 0.87
Ground-Truth Mask
1-Click Mask, Mean IoU: 0.59
5-Click Mask, Mean IoU: 0.79
10-Click Mask, Mean IoU: 0.84

Figure 18. **Additional qualitative results for point-based segmentation on the STPLS3D dataset.** For each block, we show the ground truth masks alongside our segmentation results for 1-click, 5-click, and 10-click interactions, including the corresponding mean IoU values. Points with the same color represent the same object, while clicks are highlighted using darker colors and larger spheres for better visibility.

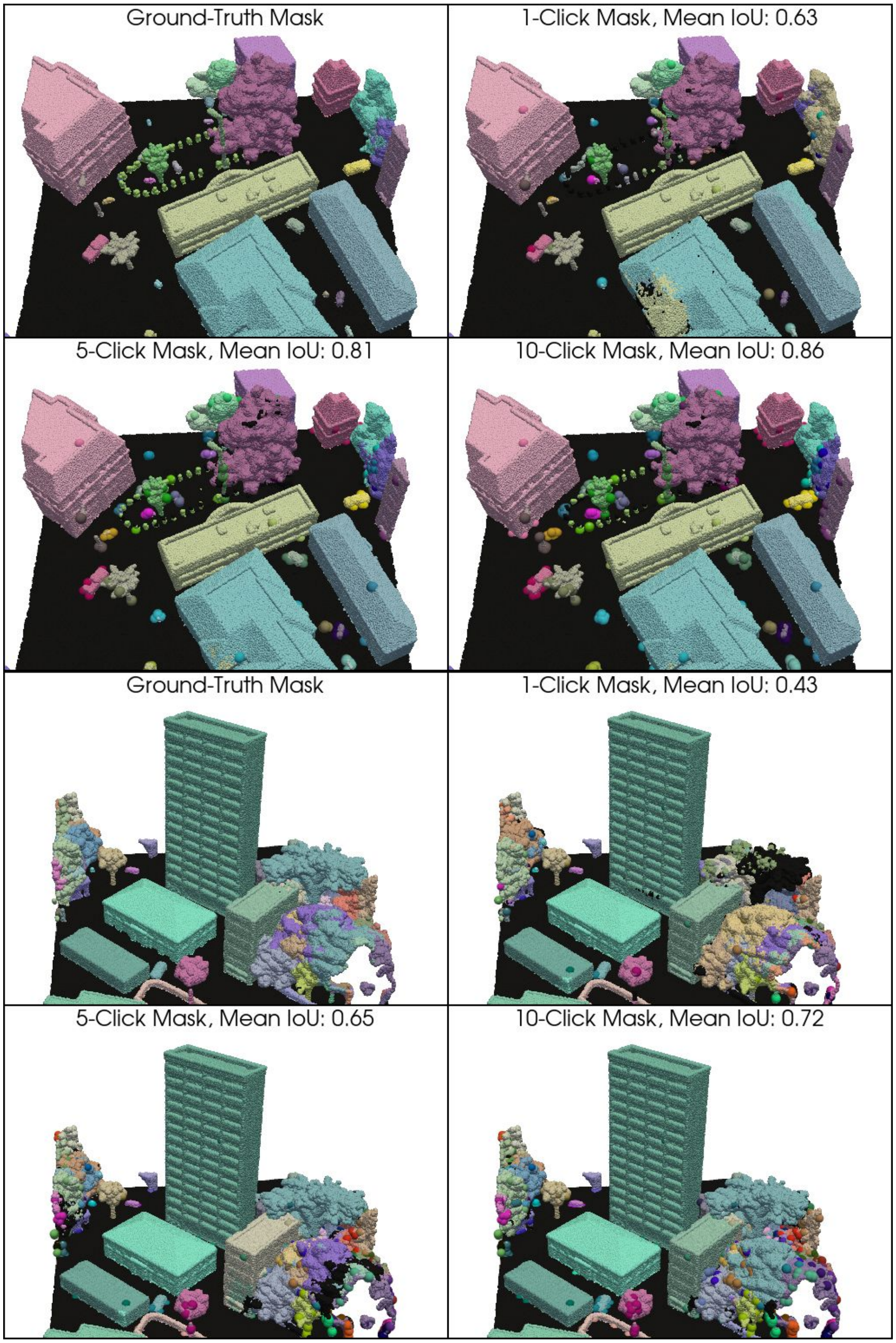

Ground-Truth Mask
1-Click Mask, Mean IoU: 0.63
5-Click Mask, Mean IoU: 0.81
10-Click Mask, Mean IoU: 0.86
Ground-Truth Mask
1-Click Mask, Mean IoU: 0.43
5-Click Mask, Mean IoU: 0.65
10-Click Mask, Mean IoU: 0.72

Figure 19. **Additional qualitative results for point-based segmentation on the STPLS3D dataset.** For each block, we show the ground truth masks alongside our segmentation results for 1-click, 5-click, and 10-click interactions, including the corresponding mean IoU values. Points with the same color represent the same object, while clicks are highlighted using darker colors and larger spheres for better visibility.

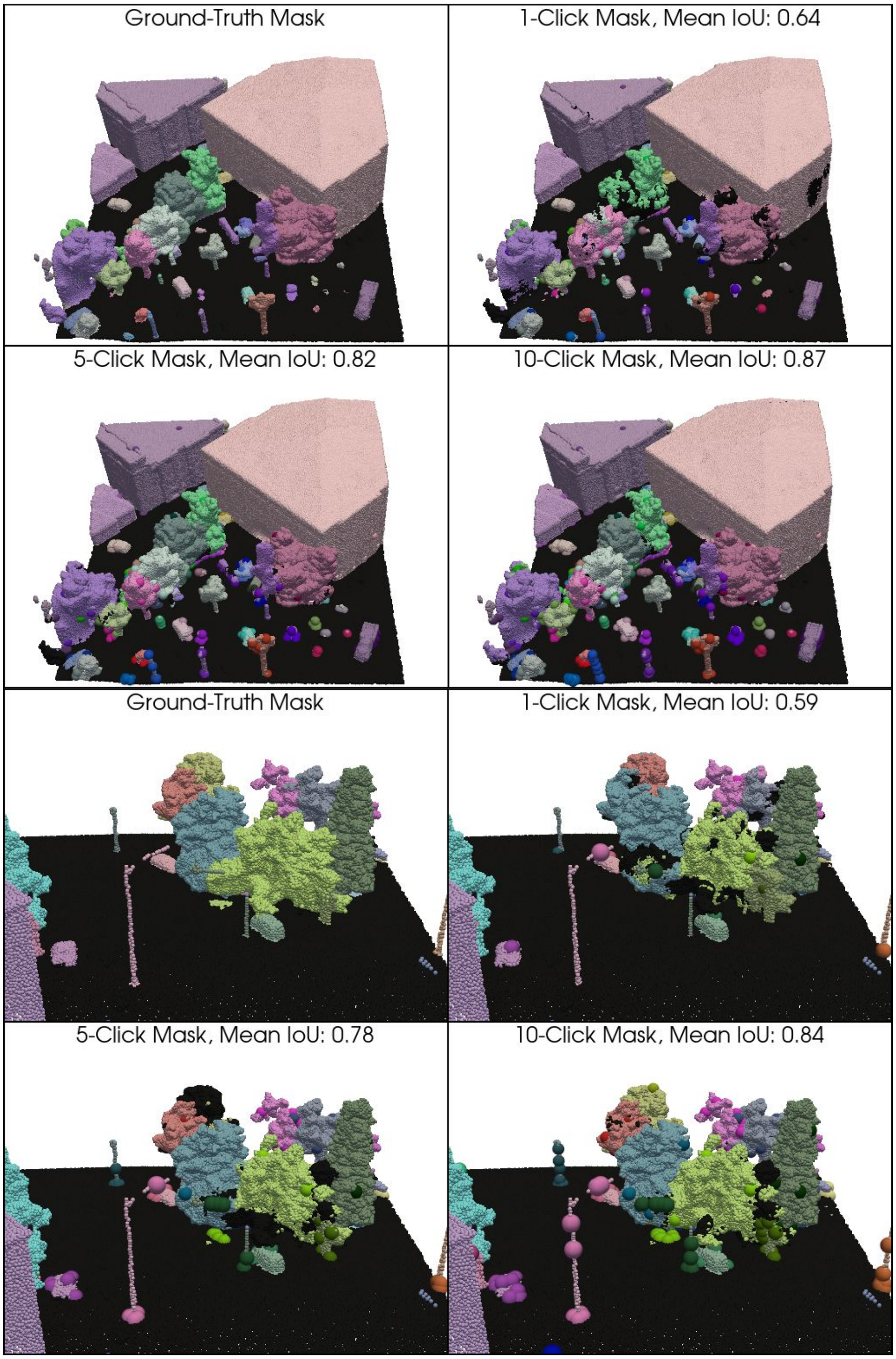

Ground-Truth Mask
1-Click Mask, Mean IoU: 0.64
5-Click Mask, Mean IoU: 0.82
10-Click Mask, Mean IoU: 0.87
Ground-Truth Mask
1-Click Mask, Mean IoU: 0.59
5-Click Mask, Mean IoU: 0.78
10-Click Mask, Mean IoU: 0.84

Figure 20. **Additional qualitative results for point-based segmentation on the STPLS3D dataset.** For each block, we show the ground truth masks alongside our segmentation results for 1-click, 5-click, and 10-click interactions, including the corresponding mean IoU values. Points with the same color represent the same object, while clicks are highlighted using darker colors and larger spheres for better visibility.